LISTING OF CLAIMS

All of the claims listed below have been allowed in the application.

1. (Previously Presented) A system comprising:

a first multiplexer associated with instruction pointers of a first thread;

a second multiplexer associated with instruction pointers of a second thread;

a first storage element dedicated to the first multiplexer; and

a second storage element dedicated to the second multiplexer, wherein

said first and second multiplexers to provide said instruction pointers of said first and second threads for execution in said processor;

one of the first and second threads is to be active while the other of said first and second threads is inactive; and

said instruction pointers for the active thread are to be delivered to processor logic;

if the first thread is inactive, said instruction pointers for the first thread are to be delivered to the first storage element for delivery to the processor logic if the first thread becomes active; and

if the second thread is inactive, said instruction pointers for the second thread are to be delivered to the second storage element for delivery to the processor logic if the second thread becomes active.

2. (Original) The system of claim 1, further comprising a common multiplexer coupled between said first and second multiplexer and processor logic.

3. (Previously Presented) The system of claim 2, wherein the common multiplexer is to receive instruction pointer data sequentially from the first multiplexer and the second multiplexer by utilizing a time-multiplexing protocol.

4. (Original) The system of claim 3, wherein the time-multiplexing protocol is a 'round-robin' protocol.

5. (Original) The system of claim 1, wherein the first multiplexer and the second multiplexer are priority multiplexers.

6. (Previously Presented) The system of claim 5, wherein the first multiplexer and the second multiplexer are to receive instruction pointer information and data from a plurality of stages in a processor pipeline.

7. (Original) The system of claim 6, wherein the first multiplexer and the second multiplexer receive instruction pointer information and data from re-steer logic at the plurality of

stages in the processor pipeline.

8. (Previously Presented) The system of claim 7, wherein the first multiplexer and the second multiplexer are to pass the instruction pointer information and data to the common multiplexer with a pre-determined priority.

9. (Original) The system of claim 1, wherein the storage element is a flip-flop device.

10. (Previously Presented) A method comprising:

associating a first multiplexer with instruction pointers of a first thread;

associating a second multiplexer with instruction pointers of a second thread;

providing, by said first and second multiplexers, said instruction pointers of said first and second threads for execution in said processor;

dedicating a first storage element to the first multiplexer; and

dedicating a second storage element to the second multiplexer, wherein

establishing one of the first and second threads as active and the other of said first and second threads as inactive;

delivering said instruction pointers for the active thread to processor logic;

if the first thread is inactive, delivering said instruction pointers for the first thread to the first storage element for delivery to the processor logic if the first thread becomes active; and

if the second thread is inactive, delivering said instruction pointers for the second thread to the second storage element for delivery to the processor logic if the second thread becomes active.

11. (Original) The method of claim 10, further comprising:

coupling a common multiplexer between said first and second multiplexer and processor logic.

12. (Original) The method of claim 11, wherein the common multiplexer receives instruction pointer data sequentially from the first multiplexer and the second multiplexer by utilizing a time-multiplexing protocol.

13. (Original) The method of claim 12, wherein the time-multiplexing protocol is a 'round-robin' protocol.

14. (Original) The method of claim 10, wherein the first multiplexer and the second multiplexer are priority multiplexers.

15. (Original) The method of claim 14, wherein the first multiplexer and the second multiplexer receive instruction pointer information and data from a plurality of stages in a

processor pipeline.

16. (Original) The method of claim 15, wherein the first multiplexer and the second multiplexer receive instruction pointer information and data from re-steer logic at the plurality of stages in the processor pipeline.

17. (Original) The method of claim 16, wherein the first multiplexer and the second multiplexer pass the instruction pointer information and data to the common multiplexer with a pre-determined priority.

18. (Original) The method of claim 10, wherein the storage element is a flip-flop device.

19. (Previously Presented) A system comprising:

a first multiplexer associated with instruction pointers of a first thread;

a second multiplexer associated with instruction pointers of a second thread;

a first storage element dedicated to the first multiplexer; and

a second storage element dedicated to the second multiplexer, wherein said first and second multiplexers provide said instruction pointers of said first and second threads for execution in said processor;

one of the first and second threads is to be active while the other of said first and second

threads is inactive;

said instruction pointers for the active thread are delivered to processor logic;

if the first thread is inactive, said instruction pointers for the first thread are to be delivered to the first storage element for delivery to the processor logic if the first thread becomes active;

if the second thread is inactive, said instruction pointers for the second thread are to be delivered to the second storage element for delivery to the processor logic if the second thread becomes active; and

a common multiplexer coupled between said first and second multiplexer and processor logic that is to receive instruction pointer data sequentially from the first multiplexer and the second multiplexer by utilizing a time-multiplexing protocol.

20. (Previously Presented) The system of claim 19, wherein the first multiplexer and the second multiplexer are to receive instruction pointer information and data from a plurality of stages in a processor pipeline.

21. (Previously Presented) The system of claim 20, wherein the first multiplexer and the second multiplexer are to receive instruction pointer information and data from re-steer logic at the plurality of stages in the processor pipeline.

22. (Original) The system of claim 19, wherein the first multiplexer and the second multiplexer pass the instruction pointer information and data to the common multiplexer with a pre-determined priority.